Hubble Space Telescope (HST)
Vision 2000 Control Center System (CCS)

System Monitoring (SYM) Subsystem

System and Functional Requirements

Prepared for

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland

September, 1996

DRAFT COPY

Abstract

This document presents the system and functional requirements for the Hubble Space Telescope (HST) Vision 2000 Control Center System (CCS) System Monitoring (SYM) Subsystem, hereafter known as SYM. SYM shall be the primary mechanism for monitoring the health and safety of the HST and the CCS to HST communications network.

> Note: The monitoring of the operational readiness of the CCS will be performed by the CCS Management (CCSM) subsystem.

Table of Contents

Preface

This is a living document and is intended to be a working description of the SYM subsystem. It is not a formal document promising delivery of the items described herein. It is intended to be a guiding influence in the design and development of the application. From that perspective, the document serves two purposes: 1) to communicate the expected functionality of the system to the Mission Engineers (MEs) and 2) to remind the developers about the needs and priorities of the end users.

The following naming conventions shall be used in this document:

Process Task	Process Name	Abbreviation
Initialize and Manage SYM tasks	SYM_ControlSym	SYM_Cnt
Manage and Distribute State Data	SYM_DistributeStateData	SYM_Dsd
Fault Detection	SYM_DetectFaults	SYM_Dtf
Fault Isolation and Recovery	SYM_RecoverFaults	SYM_Ref
Analysis & Trending	SYM_PerformAnalysisTrending	SYM_Ant
Event Management	SYM_ManageEvents	SYM_Evt
Legacy Software	SYM_PerformLegacy	SYM_Leg
DMG Interface	SYM_ManageMonitorData	SYM_Mmd

Level 1 Requirements

SYM provides eight major functions:

1.0 SYM shall provide the capability to initialize, control and synchronize SYM functions in all operational modes (SYM_ControlSym).

2.0 SYM shall provide the capability for state data distribution.. This shall include management of a local state data repository, and transmission of state data to other SYM functions and CCS subsystems as required (SYM_DistributeStateData).

3.0 SYM shall provide the capability for fault detection for the Space Network (SN) (includes NCC, STGT, JSC, and GSTDN) and the HST spacecraft, to be accomplished by tracking and comparing actual and expected conditions and reporting all anomalies (SYM_DetectFaults).

4.0 SYM shall provide the capability for fault isolation and recovery for the space network and the HST spacecraft (SYM_RecoverFaults).

Note: The Fault Isolation and Recovery requirements are provided under separate cover in the Hubble Space Telescope (HST) Vision 2000 Control Center System (CCS) System Monitoring (SYM) Subsystem, Fault Isolation and Recovery (FIR) Subsystem, System and Functional Requirements document. With the exception of the interface between the FIR functions and other SYM functions, no requirements for this subsystem will be discussed in this document

5.0 SYM shall generate Analysis & Trending products (SYM_PerformAnalysisTrending).

Note: The analysis and trending requirements are presented under separate cover in the Hubble Space Telescope (HST) Vision 2000 Control Center System (CCS) System Monitoring (SYM) Subsystem, Analysis and Trending Functional Requirements document. With the exception of the interface between the SYM_PerformAnalysisTrending function and other SYM functions, no requirements for this subsystem will be discussed in this document.

6.0 SYM shall manage CCS-generated events related to the HST or the CCS to HST communications network (SYM_ManageEvents).

Note: The CCS-generated events requirements are presented under separate cover in the Hubble Space Telescope (HST) Vision 2000 Control Center System (CCS) System Monitoring (SYM) Subsystem, CCS-generated SYM_ManageEvents function Functional Requirements document. With the exception of the interface between the SYM_ManageEvents function functions and other SYM functions, no requirements for this subsystem will be discussed in this document

7.0 SYM shall provide the existing required Legacy applications for the CCS (SYM_PerformLegacy).

Note: The legacy requirements are presented under separate cover in the __________ document. With the exception of the interface between the SYM_PerformLegacy function and other SYM functions, no requirements for this subsystem will be discussed in this document.

8.0 SYM shall provide an interface to the Data Management subsystem to manage archive of SYM generated products and retrieval of SYM required products (SYM_ManageMonitorData).

Note: The details of this function have not yet been defined. All products going to, or coming from, the Data Management subsystem have been noted in this document and the subdocuments listed in items 4.0-7.0 above. All references are within the functions producing or requiring the products.

Level 2 Requirements

Requirements in the remainder of this document apply to both modes of operation listed below unless explicitly stated otherwise using the given notations:

- EXT : Externally Generated Telemetry (FEP) and Commands (CMD)
    - Real-Time (actual current state)
    - Simulation (using external simulated data)
- RPL : Replay Saved Historical Telemetry and Commands
    - Automatic playback: merged telemetry
    - User-requested playback: real-time replay or merged telemetry
    - Simulation/test data and scenarios

Level 2 Requirements - SYM_ControlSym

The SYM_ControlSym shall manage initialization of other SYM functions (SYM_Dsd, SYM_Dtf, SYM_Ref, CCS SYM_Evt function). SYM_Cnt shall obtain and synchronize the replay of required historical information in replay mode. SYM_Cnt shall manage the export of command request from the SYM_RecoverFaults function. The SYM_Cnt function has three major subfunctions: SYM Initialization (SI), Manage Command Requests (MCR), and Manage Replay (MR).

1.1 The SYM Initialization function shall initialize the SYM_Dsd, SYM_Dtf, SYM_Ref, SYM_Evt functions. SI activities occur at system startup, after data dropout, and at the start of a replay.

1.2 The Manage Command Requests function shall receive all command requests from the SYM_Ref function, and determine whether or not they should be forwarded to the Command subsystem.

1.3 The Manage Replay function shall obtain all required historical products from the Data Management subsystem. MR shall synchronize the replay of telemetry, commands and events from these products to the SYM_Dsd, SYM_Dtf and SYM_Evt functions.

Detail Level Requirements - SYM_ControlSym

SYM Initialization

The following section of the document presents the functional and detail level requirements for the *SYM_ControlSym* , *SYM Initialization (SI)* function.

(EXT)
1.1.1 The SI shall initialize all SYM functions at system startup.

(Source: DLPs 4.02.02.02.01, 4.02.05.01.01, 4.03.04.01.01,
Legacy Req'ts PRS CMD109)

1.1.2 The SI shall re-initialize all SYM functions after data dropout.

(Source: DLPs 4.01.03.01.01, 4.02.02.02.01, 4.02.05.01.01, 4.03.04.01.01
Legacy Req'ts PRS CMD109)

1.1.3 The SI shall re-initialize all SYM functions on transition to a new PDB.

(Source: DLPs 4.01.03.01.01)

(RPL)
1.1.4 The SI shall initialize all SYM functions prior to replaying a new set of telemetry data.

(Source: DLPs 4.01.03.01.01)

(RPL)
1.1.4.1 The MR shall receive notification when merged telemetry data becomes available in the Data Management subsystem archive. This notification shall be an event message sent via the SYM_ManageEvents function, and shall include the start/stop times of the merged data.

(Source: DLPs 4.01.01.02.01, Legacy Req'ts DRS 610.7)

(RPL)
1.1.4.1.1 The SI shall provide the capability for the CCS User to enable/disable the automatic playback and monitoring of merged telemetry.

(Source: Top-Down Architecture)

(RPL)
1.1.4.2 The MR shall accept requests, from the CCS user, for historical data playback.

(Source: DLPs 4.01.01.02.01, Legacy Req'ts DRS 610.7)

1.1.5 The SI shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

1.1.5.1 The SI shall assign all such event messages a criticality.

1.1.5.2 The SI shall send, to the SYM_ManageEvents function, all such event messages.

Manage Command Requests

The following section of the document presents the functional and detail level requirements for the *SYM_ControlSym , Manage Command Requests (MCR)* function.

1.2.1 The MCR shall receive, from the SYM_Ref function, command requests for the Command subsystem.

(Source: Top-Down Architecture)

1.2.2 The MCR shall determine, based on system and user inputs, whether SYM_Ref generated command requests are permitted.

(Source: Top-Down Architecture)

1.2.3 The MCR shall send, to the SYM_Ref function, command responses to the Command subsystem.

(Source: Top-Down Architecture)

1.2.4 The MCR shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

1.2.4.1 The MCR shall assign all such event messages a criticality.

1.2.4.2 The MCR shall send, to the SYM_ManageEvents function, all such event messages.

Manage Replay

The following section of the document presents the functional and detail level requirements for the *SYM_ControlSym , Manage Replay (MR)* function.

(RPL)
1.3.1 The MR shall request for the desired time period, from the Data Management subsystem, the following archived Monitor Products:

- the Historical Integrated Command Schedule (HICS)
- replay (historical) telemetry data
- archived ES mnemonics/methods
- archived log of CCS-generated events
- archived derived parameter mnemonics/algorithms and delta value updates

(Source: 4.01.01.02.04, 4.01.01.03.01)

(RPL)
1.3.2 The MR shall provide notification, to the SYM_Dtf function, when the next scheduled command was to be sent.

(Source: Top-Down Architecture)

(RPL)
1.3.3 The MR shall provide notification, to the SYM_Dtf function, when the next scheduled command was sent.

(Source: Top-Down Architecture)

(RPL)
1.3.4 The MR shall provide, to the Event Manager function, replayed real-time events.

(Source: Top-Down Architecture)

(RPL)
1.3.5 The MR shall provide, to the SYM_Dsd, the replay telemetry data.

(Source: Top-Down Architecture)

(RPL)
1.3.6 Data originally staged in N minutes shall be played back at a CCS User selectable rate, from TBD to TBD times the original rate.

(Source: PORTS 2.2.1)

(RPL)
1.3.7 The MR shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

1.3.7.1 The MR shall assign all such event messages a criticality.

1.3.7.2 The MR shall send, to the SYM_ManageEvents function, all such event messages.

Level 2 Requirements - SYM_DistributeStateData

The SYM_DistributeStateData function shall maintain a local repository of state data and provide state data to other SYM functions and CCS subsystems. The SYM_Dsd function shall provide the capability to subscribe/unsubscribe to a continuous state data feed. The SYM_Dsd function shall provide the capability for a one-time request of current state information about one or more mnemonics. The SYM_Dsd function has three major subfunctions: Input State Data (ISD), Maintain Local Data Cache (MLDC), Manage State Data Requests (MSDR)

2.1 The Input State Data function shall accept processed telemetry from the FEP subsystem. ISD shall accept state data updates from the SYM_Dtf function.

2.2 The Maintain Local Data Cache function shall manage storage in, and retrieval from, a local cache of recent state data. This cache shall contain the most recent TBD minutes of state data.

2.3 The Manage State Data Requests function shall accept and process requests (from other SYM functions and CCS subsystems) for state data. The MSDR function shall support both one-time and continuous feed requests.

Detail Level Requirements - SYM_DistributeStateData

Input State Data

The following section of the document presents the functional and detail level requirements for the *SYM_DistributeStateData , Input State Data (ISD)* function.

(EXT)
2.1.1 The ISD shall receive HST telemetry, from the Front-End Processor (FEP) subsystem, in standard FEP format. This shall be changes only data.

(Source: DLPs 4.02.01.03.01, 4.02.02.02.01, 4.03.01.03.01, 4.03.02.02.01
Legacy Req'ts PORTS D2.2.9)

2.1.1.1 The telemetry shall consist of packets containing the following:

- Data Source
- Telemetry Format
- Spacecraft Time
- UTC Time
- GRT Time
- Number of Elements
- Element Array containing:
  - Mnemonic ID
  - Flags
  - Raw Value
  - EU Value

(EXT)
2.1.2 The ISD shall receive SMS telemetry, from the Front-End Processor (FEP), in standard FEP format. This shall be changes only data.

(Source: DLPs 4.02.01.03.04, 4.03.01.03.04, 4.03.02.02.01)

(RPL)
2.1.3 The ISD shall receive merged telemetry data from SYM_Cnt.

(Source: Top-Down Architecture)

2.1.4 The ISD shall receive state data from the SYM_Dtf function. State data shall include the following:

- Expected State Data
  - Expected State
  - ES Format (EU/Raw)
  - Tolerance
  - Time Stamp
  - Compare/Don't Compare Status
- Compare Status
- Derived Parameter True States
- Compare Process Active Status
- Orbital Events

(Source: Top-Down Architecture)

2.1.5 The ISD shall receive, from the SYM_PerformLegacy function via filtered events, state data information. The following interfaces shall be supported:

- FINATT (Fine Attitude) - difference values (angles or quaternians???) Attitude Reference Update (ARU) file - to be kept for a limited time only - as long as it is "useful"
- CRSATT (Course Attitude) - difference values (angles or quaternians???)
- DRB(Gyro Drift Rate Bias) - 3 coefficients

(Source: DLPs 4.01.01.07.01 - 05, 4.01.01.07.12 - 15, 4.01.01.08.12 - 14,
4.02.03.02.01 - 02)

2.1.6 The ISD shall receive, from the SYM_ManageEvents function, sfiltered events that contain state updates. In particular, The ISD shall receive FEP events pertinent to the actual state of the SN.

(Source: DLPs 4.02.04.03.03 - 06, 4.02.04.02.12, 4.03.03.02.03 - 06,
4.03.03.02.12, Legacy Req'ts PORTS AP417.6)

2.1.7 The ISD shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

2.1.7.1 The ISD shall assign all such event messages a criticality.

2.1.7.2 The ISD shall send, to the SYM_ManageEvents function, all such event messages.

Maintain Local Data Cache

The following section of the document presents the functional and detail level requirements for the *SYM_DistributeStateData , Maintain Local Data Cache(MLDC)* function.

2.2.1 The MLDC shall store all received state data in a local cache.

(Source: Top-Down Architecture)

2.2.2 The MLDC shall retain only the most recent TBD minutes of state data.

(Source: Legacy Req'ts PRS TLM107)

2.2.3 The MLDC shall have the capability to retrieve state data from the local cache.

(Source: Top-Down Architecture)

2.2.4 The MLDC shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

2.2.4.1 The MLDC shall assign all such event messages a criticality.

2.2.4.2 The MLDC shall send, to the SYM_ManageEvents function, all such event messages.

Manage State Data Requests

The following section of the document presents the functional and detail level requirements for the *SYM_DistributeStateData , Manage State Data Requests (MSDR)* function.

2.3.1 The MSDR shall process requests for a single snapshot of current state data for one or more mnemonics. All relevant ES, TS and comparison data shall be sent to the requester.

(Source: DLPs 4.02.02.01.07)

2.3.2 The MSDR shall process requests for current true state values. This value shall be supplied to the requester.

(Source: DLPs 3.01.02.03.02, 3.01.02.03.04, 3.01.02.04.05, 3.01.02.06.02
3.01.02.06.07,.01.03.06.01 - 02, 4.02.02.01.07, 4.02.02.04.01,
4.02.04.04.01, 4.02.04.04.03, 4.03.03.04.01, 4.03.03.04.03)

2.3.3 The MSDR shall provide the capability to allow other CCS subsystems to subscribe/unsubscribe to a continuous feed of state data information. In addition, requesters may add mnemonics to, or delete mnemonics from, their subscription list.

(Source: DLPs 4.01.01.02.11, 4.01.01.05.04, 4.01.01.09.05,
4.02.02.05.01 - 02, 4.02.04.05.01 - 02, 4.03.02.01.14,
4.03.02.02.02, 4.03.02.05.01 - 02, 4.03.03.05.01 - 02,
Legacy Req'ts PRS DSP313,314)

2.3.4 The MSDR shall process requests for comparison results (is TS = ES ± tolerance?). MSDR shall provide a Yes/No response.

(Source: DLPs 4.02.02.01.07, 4.02.02.04.01, 4.02.04.04.01, 4.02.04.04.04,
4.03.03.04.01, 4.03.03.04.04)

2.3.5 The MSDR shall process requests for recent historical state data from the local cache.

(Source: Top-Down Architecture)

2.3.6 The MSDR shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

2.3.6.1 The MSDR shall assign all such event messages a criticality.

2.3.6.2 The MSDR shall send, to the SYM_ManageEvents function, all such event messages.

Level 2 Requirements - SYM_DetectFaults

The SYM_DetectFaults subsystem shall determine the expected (predicted) and the true (actual) states of the HST and the SN, and compare these states to detect anomalies. The SYM_Dtf subsystem has three major functions: Generate Expected States (GES), Generate True States (GTS), and Manage States(MS).

3.1 The Generate Expected States function shall use information about commands (real time and stored), HST orbital events, limits and other constraints, and other information as required to determine the expected state of the HST and the SN. The GES function shall manage synchronization and updates to the expected state. The GES function shall send expected state data to the MS and SYM_Dsd functions.

3.2 The Generate True States function shall accept telemetry (real-time, replay or merged) from the SYM_Dsd function to determine the true state of the HST and the SN. The GTS function shall send true state data to the MS Function.

3.3 The Manage States function shall accept and manage expected and true state from the GES and GTS functions, respectively. The MS function shall compare expected and true states to detect miscompares(anomalies) and recoveries (previous miscompares that are no longer true). The MS function shall report all miscompares and recoveries to the SYM_Ref subsystem.

Detail Level Requirements - SYM_DetectFaults

Generate Expected States

The following section of the document presents the functional and detail level requirements for the *SYM_DetectFaults, Generate Expected States (GES)* function.

(EXT)
3.1.1 The GES shall receive, from the Command Subsystem, notification of commands sent to the HST.

(Source: DLPs 3.01.02.02.01, 3.01.02.06.08, 4.02.02.01.01 - .02
4.02.02.01.05,4.03.02.01.01 - 02, 4.03.02.01.06)

3.1.1.1 The GES shall receive notification from the Command Subsystem when the next scheduled command is ready to be sent. Required parameters may be included in this notification.

3.1.1.2 The GES shall read additional parameter information from the Integrated Command Schedule (ICS) table or other supporting data source as needed. The ICS shall include pre-planned commands and orbital events, real-time commands, and ground system configuration events ICS.

3.1.1.3 The GES shall receive, from the Command Subsystem, notification that a command has been sent to the HST.

(EXT)
3.1.2 The GES shall receive, from the Command subsystem via filtered events, notification of SN configuration changes.

(Source: DLPs 3.01.02.01.09)

(RPL)
3.1.3 The GES shall have available, from the SYM_Cnt, a Historical Integrated Command Schedule (HICS). This table shall contain the following:

- commands successfully issued in real-time
- spacecraft events

(Source: Top-Down Architecture)

(RPL)
3.1.4 The GES shall receive, from the SYM_Cnt, notification of commands sent to the HST.

(Source: Top-Down Architecture)

3.1.4.1 The GES shall receive notification from the SYM_Cnt when the next scheduled command was to be sent.

3.1.4.2 The GES shall read additional parameter information from the HICS table or other supporting data source as needed.

3.1.4.3 The GES shall receive, from the SYM_Cnt, notification that a command was sent to the HST.

(RPL)
3.1.5 The GES shall have available, from the SYM_Cnt, an archive record of mnemonic/method updates.

(Source: Top-Down Architecture)

3.1.6 The GES shall read CCS tables as necessary for required information (e.g., TLM formats, limit information, etc.) The CCS tables are:

- Project Reference Data (PRD)
- TBD

(Source: DLPs 4.02.02.01.08, 4.03.02.01.07)

3.1.7 The GES shall receive, from the SYM_ManageEvents function, events generated by CCS subsystems that affect the expected state of the HST or the SN.

(Source: Top-Down Architecture)

(RPL - User Requested Replays Only)
3.1.8 The GES shall receive, from the CCS User, requests to define ES generation methods.

(Source: Top-Down Architecture)

3.1.8.1 The GES shall provide the capability for the CCS User to define mnemonics with their associated ES generation methods.

3.1.8.2 The GES shall provide the capability for the CCS User to define/modify ES generation methods for actual (non-derived) telemetry.

3.1.9 The GES shall determine, by user-defined means (e.g., table lookup, neural net, inference engine, numerical calculation, external function, etc.), changes to expected HST or SN states.

(Source: DLPs 4.02.02.01.06, 4.02.02.01.10 - 12, 4.02.02.03.03,
4.03.02.01.03, 4.03.02.01.08 - 10)

3.1.9.1 The GES shall maintain a table of commands and orbital event directives with associated ES generation specifications.

3.1.9.2 The GES shall determine expected states for safemodes (inertial hold, software sunpoint, zero gyro s/w sunpoint, reduced power s/w sunpoint, spin stabilized sunpoint, hardware sunpoint, gravity gradient).

3.1.9.3 The GES shall maintain a time-stamped, time-ordered listing of expected states (this time is the ES time, not the command/event time).

3.1.9.4 The GES shall determine which individual expected states are in a don’t compare state (i.e., they should not be compared).

3.1.9.5 The GES shall determine when no true state data is expected (e.g., ZOEs) and therefore the MS compare function should be stopped.

3.1.10 The GES shall manage synchronization and timing of expected state verification. The GES shall send any required timing to the MS function.

(Source: Top-Down Architecture)

3.1.10.1 The GES shall determine the start of the time interval for expected states to be met(minimum time to wait).

3.1.10.2 The GES shall determine the end of the time interval for expected states to be met (maximum time to wait).

3.1.11 The GES shall send expected states to the Manage States and SYM_DistributeStateData functions.

(Source: DLPs 4.01.01.02.06, 4.02.02.01.13 - 14, 4.02.02.03.01,
4.03.02.01.11 - 12)

3.1.11.1 The GES shall assemble ES data to be sent.

3.1.11.2 The GES shall send the assembled ES data. Expected state data shall include the following:

- Expected State
- ES Format (EU/Raw)
- Tolerance
- Time Stamp
- Compare/Don’t Compare Status

3.1.12 The GES shall send to the Manage States function, directives to stop and start the compare function. These directives are based on when true state data is not expected, or when the expected state cannot otherwise be generated.

(Source: DLPs 4.02.02.03.03)

3.1.13 The GES shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

3.1.13.1 The GES shall assign all such event messages a criticality.

3.1.13.2 The GES shall send, to the SYM_ManageEvents function, all such event messages.

Generate True States

The following section of the document presents the functional and detail level requirements for the *SYM_DetectFaults, Generate True States (GTS)* function.

(EXT)
3.2.1 The GTS shall receive HST telemetry, from the SYM_Dsd, in standard FEP format. This shall be changes only data.

(Source: Top-Down Architecture)

(EXT)
3.2.2 The GTS shall receive SMS telemetry, from the SYM_Dsd, in standard FEP format. This shall be changes only data.

(Source: Top-Down Architecture)

3.2.3 The GTS shall receive, from the SYM_Dsd function, the current state of the SN.

(Source: DLPs 4.02.04.03.03 - 06, 4.02.04.02.12, 4.03.03.02.03 - 06,
4.03.03.02.12, Legacy Req'ts PORTS AP417.6)

(RPL)
3.2.4 The GTS shall receive, from the SYM_Dsd, merged telemetry data (includes HST and SMS).

(Source: Top-Down Architecture)

(RPL - User Requested Replays Only)
3.2.5 The GTS shall receive, from the CCS User, requests to define derived mnemonics and their associated algorithms.

(Source: Top-Down Architecture)

3.2.6 The GTS shall process telemetry and convert to True State (TS) data.

(Source: Top-Down Architecture)

3.2.6.1 The GTS shall have the capability to individually filter selected mnemonics to eliminate insignificant changes in value (noise). Any change in value which is less than a specified delta amount shall be discarded; therefore, only significant changes in data will be kept for further analysis.

(RPL - User Requested Replays Only)
3.2.6.2 The GTS shall receive, from the CCS User, requests to modify the delta value for a specific mnemonic.

3.2.6.3 The GTS shall calculate derived parameter values.

3.2.7 The GTS shall send true states to the Manage States function for verification.

(Source: DLPs 4.02.02.02.02, 4.03.02.02.03)

3.2.7.1 The GTS shall assemble TS data to be sent to the Manage States function.

3.2.7.2 The GTS shall send, to the Manage States function, TS data. True state data shall include the following:

- True State - Raw Value
- True State - EU Value
- Flags
- Time Stamp

3.2.8 The GTS shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

3.2.8.1 The GTS shall assign all such event messages a criticality.

3.2.8.2 The GTS shall send, to the SYM_ManageEvents function, all such event messages.

Manage States

The following section of the document presents the functional and detail level requirements for the *SYM_DetectFaults, Manage States (MS)* function.

3.3.1 The MS shall receive expected state data from the GES function. Expected state data shall include the following:

- Expected State
- ES Format (EU/Raw)
- Range Delta
- Time Stamp
- Care/Don’t Care Status

(Source: DLPs 4.02.02.01.06, 4.02.02.01.14, 4.03.02.03.01)

3.3.2 The MS shall receive true state data from the GTS function. True state data shall include the following:

- True State - Raw Value
- True State - EU Value
- Flags
- Time Stamp

(Source: DLPs 4.01.01.04.02, 4.02.02.02.02, 4.03.02.02.03, 4.03.02.03.02)

3.3.3 The MS shall maintain a table of current true and expected states. All state data received shall be entered into this table.

(Source: DLPs 4.02.02.01.14, 4.02.02.02.02, 4.03.02.02.03,
4.03.02.03.01 - 02)

3.3.3.1 The MS shall retain a recent history of state data for use by the SYM_Ref subsystem.

3.3.4 The MS shall compare true and expected states for each mnemonic as specified.

(Source: DLPs 4.01.01.05.01 - 02, 4.01.01.06.11 - 12, 4.02.02.01.09,
4.02.02.03.01 - 04, 4.02.04.03.01, 4.03.02.03.02 - 03,
4.03.03.03.01, Legacy Req'ts PRS TLM 120)

3.3.4.1 The MS shall perform the following comparison for each mnemonic: True State = Expected State ± Delta? The MS shall maintain the results of these comparisons .

3.3.4.2 The MS shall generate a "miscompare" whenever a mnemonic's true state does not fall within its allowed range.

3.3.4.3 The MS shall assign a criticality to each miscompare.

3.3.4.4 The MS shall generate a "recovery" whenever a mnemonic's true state re-enters a valid state from a miscompare state.

3.3.4.5 The MS shall receive, from the Generate Expected States function, directives to stop and start the compare engine.

3.3.5 The MS shall send all detected miscompares and recoveries to the SYM_ManageEvents function. Each text message shall include the following:

- mnemonic ID
- TS value
- ES value
- criticality
- miscompare/recovery flag
- miscompare/recovery time of occurrence

(Source: DLPs 4.01.01.05.03, 4.01.03.01.03 - 04, 4.02.02.03.06,
4.02.04.03.02, 4.02.05.01.03 - 04, 4.03.03.03.02,
4.03.04.01.03 - 04)

3.3.6 The MS shall provide, to the SYM_Ref function, a full set of information for each miscompare/recovery reported to the SYM_ManageEvents function function. This information set shall include the following:

- mnemonic ID
- TLM format
- data source
- conversion (EU/Raw)
- TS value
- TS time stamp
- TS flags
- ES value
- ES tolerance
- ES time stamp
- criticality
- miscompare/recovery flag
- miscompare/recovery time of occurrence

(Source: DLPs 4.01.01.05.03, 4.01.03.01.03 - 04, 4.02.02.03.06,
4.02.05.01.03 - 04, 4.03.04.01.03 - 04)

3.3.7 The MS shall notify the SYM_ManageEvents function when the compare engine is stopped or started by the GES function.

(Source: DLPs 4.01.03.01.05 4.02.02.03.05, 4.02.05.01.05, 4.03.04.01.05)

3.3.8 The MS shall generate event messages for all significant failures and status updates.

(Source: Top-Down Architecture)

3.3.8.1 The MS shall assign all such event messages a criticality.

3.3.8.2 The MS shall send, to the SYM_ManageEvents function, all such event messages.

Issues

The following issues are still under discussion:

- What mechanism will be used to update Expected State methods and rules, and derived parameters? If this is done via PDB updates, no interim info for ES methods, DP calculations, and Deltas is needed.
- How/where will the detailed/debug logs for testing be handled?
- Some Legacy Software issues remain unresolved; specifically, how will the Legacy software be triggered? Also, is there any Legacy Software, other than that already addressed in this document, which will involve SYM?
- For Space Network Expected States, will commanding provide notification of upcoming network configuration changes before they are requested (like they will do with HST commands)?